

CS 179N Project Raven Write Up

Darren Houn
Emmilio Segovia
Lorenzo Alamillo
Danny Diep
Chris Hyun

June 3, 2018

1 Assets

This section lists all assets used in the game, where it came from, and how it was made.

1.1 Art

- Artist: Emmilio Segovia
 - Menus and User Interface
 - Guard Rails
 - Title Screen and Dialogue Box
 - Reception Desk
 - Stands
 - Plants
 - Conveyor Belt
- Artist: Chris Hyun
 - Characters and Animations
 - * Daun
 - * Ryn
 - * Lalo
 - * Milovia
 - * Player Character
 - Creatures and Animations
 - * Beryl
 - * Borys
 - * Lulu
 - * Ner
 - * Sana
 - Objects
 - * Baggage Claim
 - * Baggage Key
 - * Key
 - * Luggage
 - * Metal Detector
 - * Note

- * Sign Post
- * Switches
- * Pillar
- * Clock
- Artist: Darren Houn
 - Airport
 - Airport Floor
 - Water Fountain
 - Boulder
 - Heated Boulder
 - Horizontal Sleeping Bag (Open/Closed)
 - Lamp
 - Potted Plants
 - Vertical Sleeping Bag

1.2 Music

- Freesound, Licenses: Creative Commons zero, by, and by-nc, Link to Licenses: <https://freesound.org/help/faq/#freesound>
- Free Music Archive, License: FMA License , Link: https://freemusicarchive.org/FMA_Licenselegalcode

1.3 Software

Tools and libraries used and their licenses.

- Unity Engine, License: Proprietary, Link to EULA: <https://unity3d.com/legal/terms-of-service>
- LightJson, License: MIT, Link to Github: <https://github.com/MarcosLopezC/LightJson>
- Piskel, License: Apache License 2.0 Link to Github: <https://github.com/piskelapp/piskel/blob/master/LICENSE>
- Krita, License: GNU GPL v3, Link to License: <https://krita.org/en/download/krita-desktop/>

2 Objectives

This section will state which objectives we have accomplished, which objectives we have dropped, what objectives we were unable to complete and the reasons why. If something is implemented but not quite working, tell me about it.

2.1 What Was Accomplished

- Technical Accomplishments
 - Character Movement - Simple omni-directional movement for the main character
 - Camera Movement - Camera moves in cardinal direction in set change of length. Smooth transition has been implemented.
 - Events - Interactions with certain objects and collision triggers will trigger events which will allow further advancement.
 - Dialogue - Interacting with other creatures and character will trigger dialogue between the player and other characters.
 - Time - Implemented a system where time passes and accounted for in game.

- Interactions - Implemented a system where the player can interact with objects and other characters.
- Save and Load - Saving the game and loading it in the future.
- Puzzles implemented and functional.
- Creative Accomplishments
 - Character sprites drawn and fully animated.
 - Creature sprites drawn and fully animated.
 - Environmental object sprites drawn and some animated.
 - Tile maps created for Airport setting.
 - User Interface designed and implemented.

2.2 What Was Dropped

- Checkpoints - They were deemed unnecessary due to the fact that the player is allowed to save their progress at any point in the game.
- Extra Environmental Art - Games have always been bottlenecked but the amount of art assets produced by the team. Art takes extensive lengths of time to complete and even longer to perfect.
- Future Levels - They have not been implemented because of need of time for extra pixel art that we cannot make in time. The pixel art needed may take even weeks to complete and we overestimated our art skills, but the design for the future levels are complete.

3 Things to Improve On

3.1 What Went Wrong

We initially factored in the time to learn a new game engine when designing the project proposal so we went with a simple game design. It was only when we started working on it that we discovered that this type of game, the good ones at least have lots of variety in interactions, event tracking, etc. What was initially assumed to be a simple game to implement became difficult when writing out extensive story dialogue. We adhered to good programming guidelines to make clean, extensible code so that if we return to this project in the future, we can easily understand the layout of the files. The core systems of game play implemented were robust but we lacked the time to implement the different details and special code needed to modify the core systems.

3.2 What We Could Do Better

One thing we could have done better is have more communication between team members because each team member was assigned jobs that gives them work to accomplish that is different than other team members. The communication starts to fail because each team member does not know what each other have done and implemented. We could have implemented a system where each team gives a overview of what they have accomplished each lab so each team member is updated on the proceedings of the project.

Another thing we could have done better is have more direction and clarity when assigning tasks to team members. The team can improve this by discussing within the group how to implement features and systems that requires other features and systems developed by another team member.